CMPSC 431W Project Guideline

This project will be a joint effort of two students in the class. Both students will be served as co-leads of the project. In this project, you choose your own domain (the thing that the project is about); this should be something that one or more of your teammates knows something about. For example, race cars, poker tournaments, vegetarian cooking, etc. Your team needs to design, implement, test, and demonstrate that your program (application) runs properly on one of the servers assigned to you this semester.

As part of the project, you will need to implement a full-fledged system with user-faced interfaces, which can be a CLI (command-line interface), website, or desktop application. All user features proposed for your project (see below) need to have an explicit public interface in your system. In the end, you will be documenting all public interfaces in the system and demonstrating the use of your product to the course staff.

Here are the details:

- Your project needs to have **at least eight tables**. You will be supplying a data model, and **it must be in BCNF**.
- You will need to **define keys and indexes** as if your product were in use for thousands of instances/records. You only need to demonstrate the product with a small subset of realistic data. However, more points will be awarded if you include a relatively large data set. This might be generated by you using a program or obtained from a published data set that is available and open for general public use. Be sure not to violate any intellectual property laws, and you must disclose the source of any data sets you use.
- There must be at least one query that **joins at least five of the tables in a useful way**. (A "non-useful way" would be if you just added two otherwise useless tables to a query just to say it joined five tables.) This query may be a report from your product (e.g., a list of all the food items, quantities, and sources required in a vegetarian recipe).
- Your application has to be backed by a **traditional server-client RDBMS**. Common open-source options are MariaDB, MySQL, PostgreSQL, etc. You should not use an embedded database like SQLite or DuckDB to host your database.
- Your application must interact with the underlying database system through **explicit SQL queries**. Thus, the usage of **ORM tools or other high-level frameworks** is **NOT** allowed.
- Your application must have at least 10 user-level features, which must include at least one instance from each of the following categories:
    - A way of inserting data
    - A way of deleting data
    - A way of updating data
    - An analytical report that must include a sorting option.
    - A transaction that involves more than a single SQL statement with at least one justifiable rollback.

- Your application must include **error trapping and recovery**. For example, if a certain value is out of the predefined value domain (bank balance < 0), your system should not panic and crash, rather it should catch and handle it in a benign manner as if it did not happen, as well as provide useful feedback to the users about the error.
- An example application can be: Considering a management system for a restaurant. A viable system should at least provide an employee login, add/modify/delete an employee to a shift schedule, print out the shift schedule for the week, add an item to the menu, change the price of an item on the menu, print out a full menu, print out an allergen menu, enter food orders for each table, calculate the bill for a table.

Please note that this project is relatively free-form, leaving you and your teammate the opportunity to think and implement creatively. Since this project is open, there should be ***NO substantive duplication of ideas or code across multiple projects***. Any such duplication will be investigated for academic integrity violations. In addition, please keep in mind that ***"borrowing" your project code from the internet is grounds for obtaining an "F"*** and being referred to the academic integrity committee. To be clear, you are certainly allowed to look up syntax and small code snippets and use them with proper citation. However, you are not allowed to use someone else's completed or semi-completed project. In this regard, note that I reserve the right to investigate and search for such issues. In short, ***do the work yourselves and you will be fine!***

The project assessment will be a collection of documents, source code, and a short demo. In particular, you and your teammate will need to submit the following material:

- **Stage I:** (1) *A requirement document with the high-level application description (you should not include any SQL code in this document) and all desired features*. It should include the usage background of your system, a list of features (at least 10 as mentioned above) with their requirements, and a description of data (on its content, semantics, and format) that will be collected and managed by the system. (2) *A signed team contract* for the agreement on work distribution of the project between two co-leaders.
- **Stage II**: *A database design document based on your requirement document finished in Stage I*. It should cover the whole conceptual database design procedures including ER diagram design, database normalization, and fully spelled-out SQL statements including all DDL/DML for all your database tables and to support your proposed features.
- **Stage III**: (1) *A user manual* with detailed instructions for system users to use all the public interfaces of your system, and instructions for developers to deploy your application from scratch. (2) *Your source code* is submitted through a private GitHub repo. (3) *A short demonstration* for the course staff of your system. (4) *A short final report examines the effectiveness of the execution of your team contract*.